**Why is this sentence complex? Cherry-pick the optimal set of features!** (Dominique Brunato & Giulia Venturi)

Institute for Computational Linguistics “A. Zampolli” (ILC-CNR), ItaliaNLP Lab, Pisa

Description of the method

The methodology we propose is inspired to research on linguistic profiling (van Halteren, 2004) carried out on automatically or manually (gold) annotated corpora. It takes into account 11 linguistic features, which have been selected as metrics identifying factors of complexity at different levels. All features are extracted from single sentences, which we consider as our units of analysis, with a main focus on the syntactic level.

The method was applied to all the provided UD treebanks. For each feature extracted from sentence, we calculated the average value it has in all sentences contained in each treebank. In order to compare features that have different scales (e.g. percentage, absolute number), for each feature we scaled its average value in the range of all the average values that the feature has in all treebanks. We scaled each feature in a range between the minimum value and 1.0, which corresponds to the maximum value in that range.

According to the literature on sentence complexity from different perspectives (cognitive, corpus-based, computational), we assumed that the higher the value, the more complex is the language usage described in the treebank with respect to each feature.

Here follows the description of features, corresponding to the metrics reported in the csv:

1. Sentence length (“BV_n_tokens” in csv): it is calculated as the average number of words per sentence. Sentence length is typically used as an approximation of syntactic complexity, for example in traditional formulas developed for the automatic assessment of text readability.

2. Word length (“BV_char_per_tok” in csv): it is calculated as the average number of characters per word (excluded punctuation). It is a basic indicator of word complexity and, similarly to sentence length, it is used by traditional readability formulas as an approximation of lexical complexity.

3. Distribution of verbal head (“BV_verbal_head_per_sent” in csv): it computes the average number of verbal heads in a sentence. Verbal heads are indicative of the presence of (independent and/or dependent) clauses, thus the higher their distribution the more complex the sentence could be. However, the main drawback is that also the opposite trend could be a symptom of complexity, since the information explicitly provided by the verb could be conveyed e.g. by elliptical constructions or nominalizations.

4. Distribution of verbal roots (“BV_verbal_root_perc” in csv): it calculates the average percentage of roots headed by a lemma tagged as verb out of the total of sentence roots. Similarly to 3, a lower distribution of verbal roots might suggest a higher presence of nominal or copular sentences.

5. Clause length (“BV_avg_token_per_clause” in csv): it is measured as the number of tokens occurring within a clause, which is here calculated as the ratio between the number of verbal heads to the total number of tokens. Note that this operationalization is a proxy of the identification of clause length since we do not consider the subtree of nominal predicates. Syntactic metrics relying

on clause length, such as T-Unit (Hunt, 1966), are widely used e.g. in first and second language acquisition to assess the development of syntactic competence.

6. Length of dependency links (“BV_avg_links_len” in csv): it is calculated as the average number of words occurring between the syntactic head and the dependent. In psycholinguistics studies, it is a well-known factor used to explain how syntax impacts human sentence processing (Gibson 1998, 2000), i.e. the longer the dependency, the higher the cognitive load which is needed to derive a meaningful representation of the sentence; from an NLP perspective, this metric is also used as an explanation of lower performance by statistical parsers (Rimell et al., 2009; Nivre et al., 2010).

7. Depth of the whole parse tree (“BV_avg_max_depth” in csv): it corresponds to the longest path from the root of the dependency tree to some leaf. It has been shown that deeper syntactic structure negatively affect human sentence processing (Frazier, 1985).

8. Verb arity (“BV_avg_verb_edges” in csv): it corresponds to the average number of instantiated dependency links (both arguments and modifiers) sharing the same verbal head, excluding auxiliaries bearing the syntactic role of copula according to the UD scheme. This feature reflects the richness of verbal predicates, i.e the higher the score the richer the verbal predicate. Note that this measure could be calculated in a more sophisticated way if corpora had a further level of annotation making explicit the verb argument structure (allowing to distinguish arguments from adjuncts) or an external subcategorization lexicon which can be used as a reference resource;

9. Average depth of ‘chains’ of embedded subordinate clauses (“BV_avg_subordinate_chain_len” in csv): features 9, 10 and 11 model syntactic phenomena correlated to the use of subordination, a broadly studied marker of structural complexity. Note that all these features do not distinguish subordinate clauses in terms of types nor between finite vs non-finite clauses.

10. Distribution of subordinate clauses preceding the main clause (“BV_subordinate_pre” in csv): this feature and feature 11 refer to the relative position of subordinate clause with respect to the main clause. Note that variations of these two features should be interpreted considering the canonical order of the language and the degree of word order freedom that each language allows.

11. Distribution of subordinate clauses following the main clause (“BV_subordinate_post” in csv).